IJCSI International Journal of Computer Science Issues, Vol. 7, Issue 3, No 6, May 2010
ISSN (Online): 1694-0784
ISSN (Print): 1694-0814

Latency Optimized Data Aggregation Timing Model for Wireless Sensor Networks

A.Sivagami[1], K. Pavai[2] and D. Sridharan[3]

[1]Research Scholar, Department of Electronics and Communication Engineering, Anna University Chennai, 600 025, Tamil Nadu, India

[2]Research Scholar, Department of Electronics and Communication Engineering, Anna University Chennai, 600 025, Tamil Nadu, India

[3]Assistant Professor, Department of Electronics and Communication Engineering, Anna University Chennai, 600 025, Tamil Nadu, India

Abstract

In Wireless Sensor Networks (WSN), the data aggregation schemes are extensively used to avoid the redundant transmission of correlated data from densely deployed sensor nodes. Even though, it enhances the network lifetime, it seriously suffers from the increased data delivery time. The high end-to-end delay experienced by the packets is not acceptable in the delay-constrained applications like seismic activity monitoring, military field monitoring, etc. In this work, we propose a novel data aggregation-timing model for node's aggregation time out to reduce the data delivery time.

Keywords: *Wireless Sensor Networks, Data Aggregation, latency minimization, timing model, TOSSIM.*

1. Introduction

The recent development in the field of Micro-Electro-Mechanical Systems (MEMS), brings the dream of developing a low cost, tiny and autonomous device called wireless sensor node, capable of sensing, processing and communicating the field data, into reality. The low processing and limited communication capabilities of the nodes demand the dense deployment of sensor nodes in the monitoring terrain to cover the entire area and provide fault tolerance to node failure. The densely deployed nodes will sense the similar data and there is a high correlation among these data. It is not worthy to transmit the similar information by many nodes, since the communication cost is the dominant energy consumer in WSN. Many efforts are taken to reduce the number of unwanted transmissions in sensor networks. The data aggregation techniques gained more attention in achieving energy saving in WSN. The data aggregation is a technique to combine the data from various sensor nodes to eliminate the redundant information and provide the rich and multi- dimensional view of the monitoring environment [1]. Many data aggregation protocols were proposed in literature [2-4] to reduce the energy consumption. But the data aggregation algorithms suffer from the increased data delivery time due to the fact that the aggregator nodes have to wait for the data from its children nodes. If the aggregator waits for more time, it collects more data from its children and hence aggregation gain increases. This will increase the gain with the increased delay and vice versa. Hence, there is a trade off between the energy and the delay [5].

The Wireless Sensor Networks are mainly deployed for transferring real world data wirelessly and hence it is not worthy to transmit the information to the base station belatedly. There are many applications, which need the time stringent data delivery, pose the challenge in developing data aggregation method that guarantee the delay requirement. If the aggregator node waits long time for the data from all its children, the data delivery time to the sink increases and the data of the present round interfere with that of next round. The timing model defines how long an aggregator node should wait for the data from its children. The aggregation time out should be such a way that the waiting time should be optimum to optimize the data aggregation and deliver the data to the sink within the stipulated time bound. Some of the works to reduce the delay while aggregating data are considered here.

The rest of the paper is organized as follows. In Section 2 the related work in this domain is outlined, in the Section 3 the proposed algorithm is explained, the protocol implementation is given in the section 4, in the Section 5, the performance of the algorithm is presented and the Section 6 gives the final conclusion of the work.

2. Related Work

The aggregation time out may be *periodic simple*, that all the nodes can wait for the fixed predetermined time period or *periodic per hop*, the aggregator waits to hear from all of its children or *cascading timeout*, the time out depends on its position in the data aggregation tree (DAT). In cascading timeout [6], nodes schedule their time out based on its position in the DAT. A node's time out happens after its children's time out and thus enables a node to collect information from all its children. But, all the nodes in the particular level are given with same time out irrespective of its number of children. The advantages of this algorithm are it does not require any time synchronization or any centralized control. It doesn't consider the number of children in each tree and thus the nodes with more children will loose some of its children data and also the nodes in the same level will send the data at almost same time, leading to traffic congestion.

In [7], the nodes time out is determined dynamically based on the aggregation tree structure and the number of children it has. The node's aggregation time out may be increased when a node detects the dead line miss. The update process is more complex and the agent nodes, which are one hop neighbors to the sink, will be congested more since the data arrive at the agent node almost simultaneously.

The Adaptive Time Control (ATC) [8] determines the aggregation time out for a node based on the level of the sensor node in the data aggregation tree and the number of children it has. The nodes with more children are given with more time and thus maximizing the opportunity for data aggregation from its children. Thus the nodes in the same level will get different aggregation time out. The authors claimed that this algorithm provides higher data delivery rate and lower energy costs compared to cascading time out. In their simulation, modified IEEE 802.11 protocol is used as medium access protocol and the node's sleeping schedule is not considered. This will have the impact on the energy consumption and the delay calculation.

In [9], the time efficient data aggregation on clustered WSN is considered. The time out is calculated for each sub tree in the cluster, which is based on the packet transmission delay and the cascading delay. The performance is compared for various modulation techniques, which is the key factor for the packet transmission delay.

The above protocols are simulated either using network simulator NS2 or discrete simulator based on C++. Also, they have used IEEE802.11 as channel access mechanism with or with out some modifications. The performance of these protocols may be tested for the real time deployment of wireless sensor nodes. The wireless sensor nodes of our consideration are IRIS motes from Crossbow Technologies [10]. It is good practice to test the performance of any protocol using a simulator before it is implemented in the real world. The simulators like ns2 and other similar simulators do not reflect the real world scenario properly. Therefore we have chosen the WSN simulator TOSSIM to test the behavior of our proposed timing model. The simulator TOSSIM can simulate the program written in NesC, which is the native language for WSN mote IRIS, and the same the code can be fused into the mote with minor modifications. [11]

3. Timing Model

The objective of this work is to develop a protocol, which delivers the data to the sink with in the deadline while adapting the data aggregation methods for reducing the energy consumption. This protocol estimates the time out of each node in the tree in a distributed manner, so that the data generated by each node should be delivered to the sink before the deadline. This work aims at the target platform as the wireless sensor mote called IRIS. These motes use TinyOS operating system, which is one of the most widely adapted operating system for the resource-constrained mote network. The motes send the data to the sink using the collection tree, which is formed and maintained by the Collection Tree Protocol (CTP) [12]. The CTP protocol uses the wireless link quality between the motes as the metric to construct the tree and it is a dynamic tree. If the link quality changes, the tree structure will also change. Hence, the aggregation time out should be dynamic and updated as the tree structure changes.

In cascading timeout, when an aggregator node receives the request from the sink, it calculates its time out based on the level in the tree. The staggered time out takes place between the levels and time disarticulation between them is just a single hop delay. The nodes in the same level are having same time out and hence they try to gain access to the channel simultaneously. Therefore, the transmission of the packets by the nodes in the same level will be deferred by the MAC and the parent time out will takes place before the child. Hence, the aggregator node will miss some of the packets from its children nodes and the aggregation gain decreases. To improve the performance of the DAT, the aggregator's node time out should be assigned in such a way that it could collect more information efficiently.

In our proposed algorithm, each node calculates the initial time out for data aggregation, which is based on the hop distance from the sink and the number of children it has in its sub tree rooted from it. This initial time out will be different for each node and it increases from the leaf node to sink node. The children nodes time out first, followed by its parent and hence the data generated by the children nodes are collected, processed and forwarded by its

parent node in staggered manner. After the initial staggered timeout, the nodes follow the fixed time out of duration T, which is data generation period. This ensures that the data generated by the nodes reach the sink before the next round begins. The Fig. 1 shows the timeout model of the proposed system called Delay Efficient Aggregation Timing Model (DEATM).

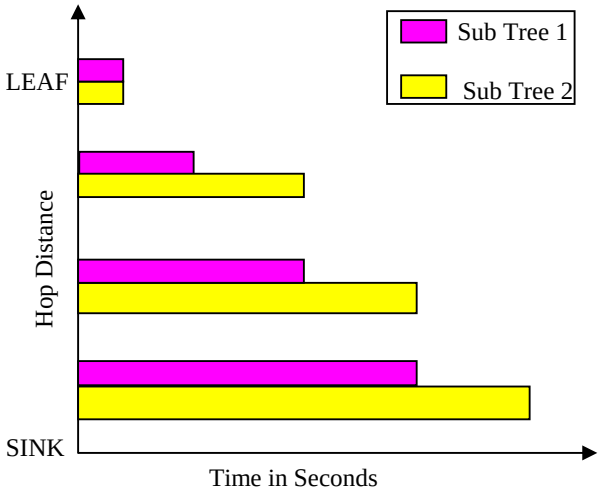

Fig. 1 Timing Model of DEATM

The Fig. 1 shows the initial time out of the nodes at different level from the sink in two different sub trees. Each color in the figure shows the aggregation timer for the nodes in the same sub tree and the figure shows two such a sub trees. The nodes in the same level are represented with different color and their initial time out will depend on the total number of children in the sub tree in which it resides. The leaf nodes get the least waiting time and the nodes near to the sink gets more staggering time. All the leaf nodes get the same staggering time but the nodes in the same higher levels will get different stagger time. The leaf node's time out takes place early and the time out of the nodes in the different levels increases as it approaches the sink. The nodes near to the sink get more time out and it is less than the deadline T. The nodes in the same level will have different time out and hence the collision in the same level be avoided and it ensures that the data wave will reaches the sink through that sub tree with in the dead line.

3.1 Mathematical Model

Let A_i is the aggregator node i and L_j is the leaf node j. Let N_i is the number of children nodes for the aggregator node A_i. i.e N_i = degree(A_i). The path cost from any leaf node L_j to an aggregator node A_i is given by

$$\text{Path\_cost}(L_j, A_i) = \sum_{A_k \in n} \deg ree(A_k) \qquad (1)$$

Where n is the set of aggregator nodes in the path from L_j to A_i. The maximum path cost from any leaf node to an aggregator node is P_i is the maximum path cost from leaf node to the aggregator.

$$P_i = \max \{\text{Path\_cost}(L_j, A_i)\} \quad \forall j \qquad (2)$$

The maximum path cost from any leaf node to sink is P_{sink} is the maximum path cost from leaf node to the aggregator.

$$P_{sink} = \max \{\text{Path\_cost}(L_j, \text{sink})\} \ \forall i, j \qquad (3)$$

The initial staggered time out is calculated as follows. Let T_i is the stagger time out for the aggregator node i, which is equal to

$$T_i = T_{ci} + T_{ai} \qquad (4)$$

Where T_{ci} is the cascading time out which depends on the level in which the aggregator is in. This gives the initial timeout as in cascade time out for each node and it is same for all the nodes in the same level. The T_{ai} is the aggregation time out of the node, which depends on the number of children it has.

$$T_{ci} = 2 * [T - (T_{TD} * h)]$$

$$T_{ai} = (P_i / P_{sink}) * (T - T_{TD} * D) \qquad (5)$$

Here, h denotes the hop distance of the node Ai, D is the depth of the tree, T is the data generation period or the dead line and T_{TD} is the one hop delay between the levels. It depends on the queuing delay, MAC delay, processing delay for aggregation function and the transmission delay. It is assumed to be 0.1 seconds as used in [8]. After introducing this initial delay in the aggregation timer, the aggregation timer is fired for every T seconds thus enabling the collection of packets generated in that round from all the nodes in the collection tree.

3.2 Update Phase

The aggregation timer is updated whenever there is a change in the topology or the aggregation gain is reduced due to time synchronization between the nodes.

The beacon messages are exchanged at regular intervals between the nodes. If there is a change in the topology, the routing engine will send the message to its neighbouring nodes and hence all the nodes in the network will be updated their information. The aggregation time out T_{ai} is recomputed every time it receives the beacon. If the new value and the previous timer value are differing more than by the given threshold Δ, then the aggregation timer will be re adjusted. Also, the optimum number of responses for each aggregator node per round is N_i, which is equal to number of children it has. If the aggregator receives less than N_i, the timer value is increased by T_{TD} and if it receives more than N_i, it reduces the time by T_{TD}.

4. Implementation

The default CTP routing protocol sets up the data collection tree by exchanging the beacon messages, which contains the information about its parent, and the cumulative link quality to reach the sink. Each node sends the beacon messages periodically. The CTP's beacon message is modified to carry the additional information about the hop distance and P_i. The neighbour table is also modified to keep the record of neighbour's hop distance and P_i. Each aggregator node finds its own children from the parent field of beacon message sent by its children. When a node receives the beacon message from its neighbor node with its node id as parent, it increments the *number of children* field in its neighbor table. Aggregator node also finds its hop distance using the beacon information from its parent. Each aggregator node calculates Path cost for each sub tree by adding the P_i of its children node and its number of children. The P_i of the node is calculated by finding the maximum value among all P_i's. This information is passed to aggregator function, which calculates the value of T_i. The initial time displacement is achieved and the data wave will reach the sink with in the stipulated time.

Each node uses two timers, one timer for the data generation and the other for the aggregation. After the completion of setting up the collection tree phase, the node starts both the timer. The data timer is fired every T seconds. The initial time out for the aggregation timer is calculated from the beacon message and the one-shot aggregation timer is started with the calculated time. When the data timer fires, the node read the default sensor and keep it in the buffer. When the aggregation timer fires first time, it restarts the aggregation timer with T seconds, which is a periodic timer. When the periodic aggregation timer fires, the node does the aggregation of the packets from its children with its own value and send the aggregated packet up in the tree. The simple aggregation operator *average* is used.

5. Simulation and Result

The proposed algorithm is simulated using TOSSIM simulator under Linux platform, which is a simulator for TinyOS2.x developed by University of California at Berekely, USA that can run the actual TinyOS code without any real motes. The 100 nodes are deployed uniformly in the area 200 × 200 m^2. The TOSSIM uses SNR based simulation, the simulator parameters are such that, it simulates the indoor environment. The nodes are placed in uniform topology, where the sensor filed is divided into grid of equal size and the node is placed randomly in each grid. The default MAC IEEE 802.15.4 is used for Channel access and CTP with four-bit link estimation as data collection tree. The nodes generate traffic for every 20 seconds.

The performance parameters of our consideration are the aggregation gain, accuracy and miss ratio. The aggregation gain is defined as the measure of reduction in the communication traffic due to aggregation, which is related to the energy of the node [13]. It is the ratio of traffic reduction due to aggregation to the total traffic without aggregation.

$$A_G = 1 - \frac{t_a}{t}$$

Where t is the number of transmissions by all nodes without aggregation. Without any aggregation, an aggregator node should forward all the packets coming from its children. Hence the value of t is calculated by adding the total number of transmissions and reception by all the nodes. t_a is the total number of transmissions with aggregation, which is equal to number of packets transmitted by all the nodes. The aggregation gain directly related to the energy consumption of the nodes and hence it has been chosen as one of the metric for analysing the protocol.

An aggregator node receives and processes N_r packets from its children, and sent the information in one packet. Without aggregation, the aggregator node has to forward all the packets irrespective of its content. Hence the total number of transmission by a node is reduced by the factor of $N_r/ (N_r +1)$. The aggregation gain is the ratio of reduction in number of transmission due to aggregation to the number of transmission without aggregation. If a node is waiting for an optimum time period, it could collect the information from all of its children, then the aggregation gain will be maximum. If an aggregator's time out is less than the optimum time out, then the aggregator node could not collect information from some of its children. Thus, these packets, which are arriving after the deadline, are not considered for finding the aggregation result and hence discarded. This will reduce the accuracy of the aggregation process. The data accuracy is a measure of how much data is used to extract the information. It is defined as the number of readings received to the total number of packets generated in the network. The aggregated packet contains how many children's data are used to find the aggregated value. Hence the sink could find the accuracy of the aggregation for a given simulation time. The accuracy of the aggregation process depends on the number of packets received from its children and in turn depends on the deadline requirement of the application and the node density of the network.

The miss ratio is another parameter to judge the efficiency of the protocol. It is defined as the ratio of number of packets missing the deadline to the number of packets received. In each data collection round, the packets of the present round arriving after the aggregation time out will be dropped. Each node will calculate the total number of packets dropped and the total number of packets it received from its children for the given simulation period. The total number of dropped and received packets by all

the nodes are taken to measure the miss ratio. These performance parameters depend on the deadline, the number of hops, the number of nodes in the sub tree and the node density in the surveillance field.

The proposed algorithm is compared with *simple per hop*, and *cascade time out*. In *simple per hop*, the nodes wait for fixed pre determined time and do the aggregation and in *cascade timeout* the staggered one hop delay time out is followed. The figure 2 to figure 7 show the results obtained from our simulation tests.

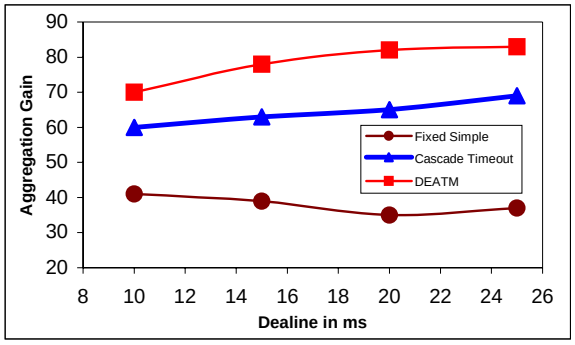

Fig. 2 Impact of Deadline on Aggregation Gain

The Fig. 2 shows the aggregation gain for the different timing models as a function of deadline. The nodes sample the sensor and generate the data for every 20 seconds. This data is aggregated with the data received from its children during the time out period and the aggregated packet is transmitted out when the aggregation timer fires. The data generation period or deadline is varied from 10 seconds to 25 seconds and the aggregation gain is measured. The aggregation gain increases as the deadline increases. If T is small, more packets will miss the deadline due to small waiting period and increased data traffic. As the deadline increase the data traffic is reduced as well as the nodes will be given more opportunities to do the aggregation and hence the gain increases. Our proposed algorithm gives better aggregation gain compared to cascade time out since in our method, the aggregator node's timeout includes the number of child nodes and hence it could collect more data from its children. In simple and cascade time out, the nodes, which are very close to the sink could send the data with in the deadline and hence the gain is less compared that of our proposed scheme.

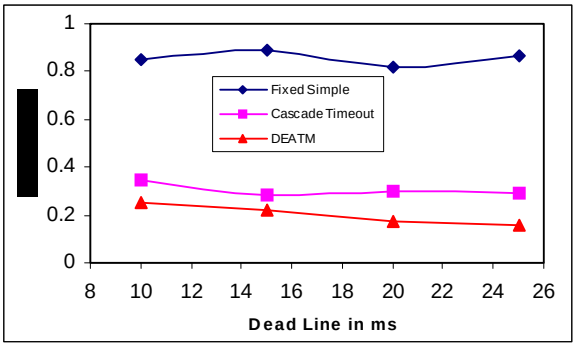

Fig. 3 Impact of Deadline on Miss Ratio

The Fig. 3 shows that the miss ratio of the proposed algorithm is very less compared to other two schemes because, each node waits appropriate time to collect the data from all its children and hence the data reaches the sink with in the deadline. As deadline increases, the miss ratio decreases.

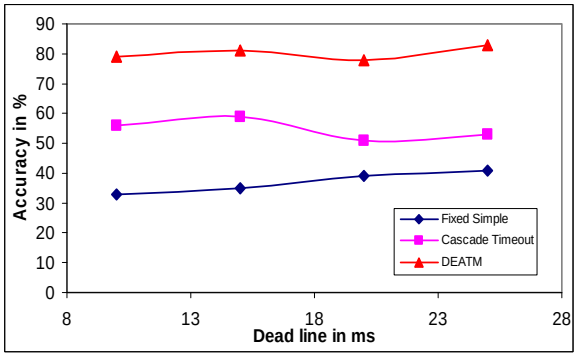

Fig. 4 Impact of Deadline on Accuracy

The fig. 4 shows the data accuracy of the proposed algorithm as a function of deadline. The accuracy of the proposed system is generally high because it could collect more data from its children within the deadline. Accuracy increases with increase in deadline.

In order to find the impact of node density on the performance parameters, the simulation is performed for the duration of 100 seconds and the number of nodes varies from 25 to 100. The area of the network is same for all the cases and hence the density of the nodes in the sensor field varies.

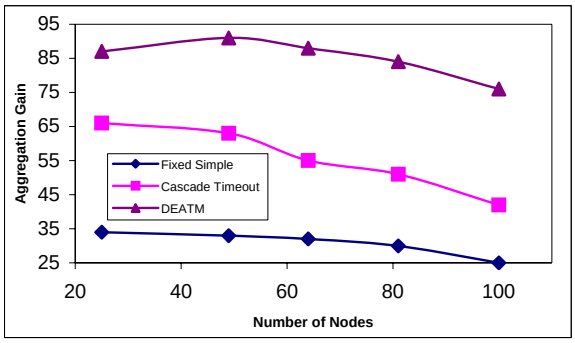

Fig. 5 Effect of Node density on Aggregation Gain

The Fig. 5 shows dependency of aggregation gain on the network size. As the network size increases, the aggregator nodes could collect more packets from its children and reduce the traffic by aggregation. Hence the gain increases as the number of nodes increases. This is not linear due to the fact that the increased number of nodes causes collision and the children nodes have to wait more time to get the channel, which leads to the decrease in gain.

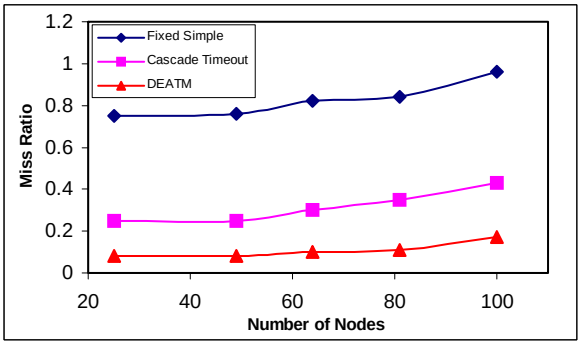

Fig. 6 Effect of Node density on Miss Ratio

The Fig. 6 shows the impact of node density on the miss ratio. The miss ratio of the proposed scheme is very less compared to the other two schemes. Also, for the small node densities, the miss ratio does not change much for all the schemes but the miss ratio increases as the node density increases. This is due to the fact that if the node density increases, the number of competitors for a node to access the channel increases. This will increase the MAC deferring time and hence the packets will miss the dead line and dropped by the aggregators.

The fig. 7 shows the data accuracy of the proposed algorithm as a function of network size. The accuracy of the proposed system is generally high and increases with decrease in network size.

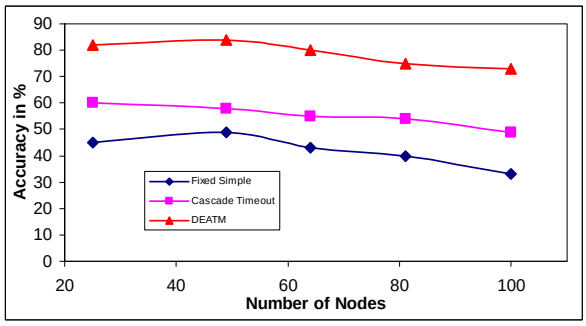

Fig. 7 Effect of Node density on Data Accuracy

From these figures, our proposed algorithm can do better than other two algorithms even in high-density networks.

5. Conclusion and Future work

The proposed protocol gives more *aggregation gain*, which leads to less energy consumption, less *miss ratio*, which delivers the data with in the stipulated time bound and the good *data accuracy*. Thus our protocol can deliver more accurate and fresh information to the sink in an energy efficient manner.

Also, our proposed algorithm is independent of time synchronization and it doesn't need any centralized control. It also adjusts the time dynamically according to the change in topology or change in synchronization. This proposed algorithm gives more aggregation gain compared to that of cascading time out scheme and the data generated in a round is delivered to the sink in the same round. Thus the data freshness is maintained.

The gain, miss ratio and the accuracy of the protocol depend on the deadline and the size of the network. From our observations, for a given network size, a minimum deadline has to be fixed so that the data can be delivered to the sink with in the deadline.

The proposed work will be tested with the real test bed consists of IRIS motes.

References

[1] R Rajagopalan and P. K. Varshney, "Data Aggregation techniques in sensor networks: A Survey", Journal on IEEE Communications, Surveys and Tutorials, Vol. 8, Issue 4, 2006, pp.48-63.

[2] Mohamed A. Sharaf, Jonathan Beaver, Alexandros Labrinidis and Panos K. Chrysanthis, "TiNA: A Scheme for Temporal Coherency Aware in Network Aggregation", in 3rd ACM International Workshop on Data Engineering for Wireless and Mobile Access, 2003, pp. 69-76.

[3] Tian He, Brian M. Blum, John A. Stankovic and Tarek Abdelzaher, "AIDA: Adaptive Application-independent Data Aggregation in wireless Sensor Networks", ACM Transactions on Embedded Computing Systems, Vol. 3, No. 2, May 2004, pp. 426–457.

[4] Yujie Zhu, Ramanuja Vedantham, Seung-Jong Partk, Raghupathy Sivakumar, "A Scalable correlation aware aggregation strategy for wireless sensor networks", Journal of Information Fusion, Elsevier publishers, Vol. 9, Issue 3, 2008, pp. 354-369.
[5] Yang Yu, Bhaskar Krishnamachari, and Viktor K. Prasanna, "Energy-latency tradeoffs for data gathering in wireless sensor networks". in IEEE INFOCOMM, 2004.
[6] Ignacio Solis and Katia Obraczka, "The impact of timing in data aggregation for sensor networks", in IEEE International Conference on Communication (ICC), 2004.
[7] Jae Young Choi, Jong Wook Lee, Kamrok Lee, Sunghyun Choi, Wook Hyun Kwon and Hong Seong Park, **"**Aggregation Time Control Algorithm for Time constrained Data Delivery in Wireless Sensor Networks**",** in IEEE Vehicular Technology Conference, 2006, Vol. 2, pp. 563 – 567.
[8] Hong Li, HongYi Yu, BaiWei Yang and Ana Liu, "Timing control for delay-constrained data aggregation in wireless sensor networks", International Journal of communication systems (Wiley), Vol. 20, issue 7, 2007, pp. 875- 887.
[9] Shan Guo Quan and Young Yong Kim, "Fast Data Aggregation Algorithm for Minimum Delay in Clustered Ubiquitous Sensor Networks", in International Conference on Convergence and Hybrid Information Technology (ICHIT), 2008, pp. 327-333.

[10] Crossbow Technologies http://www.xbow.com
[11] Philip Levis, Nelson Lee, Matt Welsh, and David Culler, **"**TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications**",** in the First ACM International Conference on Embedded Networked Sensor Systems (SenSys), 2003, pp. 126-137.
[12] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol", in the 7th ACM International Conference on Embedded Networked Sensor Systems (SenSys), 2009, pp. 1-14.
[13] U. Roedig, A. Barroso, and C. Screenan, "Determination of Aggregation Points in Wireless Sensor Networks", in the 30th EUROMICRO conference, 2004, pp. 503 – 510.

A. Sivagami received her B. E. degree in Electronics and Communication Engineering from University of Madras in the year 1990 and obtained her Master's degree in Communication Systems from Bharathiyar University in the year 2000. Currently she is doing her research in the field of Wireless Sensor Networks in Anna University, Chennai, India. Her fields of interests are Wireless Networks, Wireless Sensor Networks, Digital Communication, etc. She has published 4 papers in International conferences and 3 in International Journals

K. Pavai received her B.E. degree in Electronics and Instrumentation Engineering in the year 2004 from university of Madras and M.E. in Applied Electronics in the year 2006 from Anna University, Chennai. Currently she is doing her research in the field of Wireless Sensor Networks in Anna University, Chennai. She has 4 publications in the National conferences, 3 in International conferences and three in International Journals. Her current fields of interests are Wireless Sensor Networks, VLSI, Mobile Communication, Instrumentation, etc.

Dr. D. Sridharan received his B.Tech. Degree in Electronics Engineering and M.E. degree in Electronics Engineering from Madras Institute of Technology, Anna University in the years 1991 and 1993 respectively. He got his Ph.D degree in the Faculty of Information and Communication Engineering, Anna University in 2005. He is currently working as Assistant Professor in the Department of Electronics and Communication Engineering, CEG Campus, Anna University, Chennai, India. His present research interests include Internet Technology, Network Security, Distributed Computing and VLSI for wireless Communications. He has published more than 25 papers in National/International Conferences and Journals.